УДК 004

Метод обнаружения сетевых атак с использованием двухэтапного анализа

Method of network attacks detection using a two-stage analysis

Колесников / Kolesnikov N.
Никита Дмитриевич
(nik-kron@mail.ru)
ФГАОУ ВО «Санкт-Петербургский национальный исследовательский университет информационных технологий механики и оптики»,
аспирант факультета безопасности информационных технологий.
г. Санкт-Петербург

Ключевые слова: обнаружение сетевых атак – network attack detection; двухэтапный анализ – two-stage analysis; анализ сетевого трафика – network traffic analysis; CICIDS2017; предобработка сетевого трафика – network traffic preprocessing; смешанные сети – mixed networks; информационная безопасность – information security.

В работе представлен метод обнаружения сетевых атак с использованием двухэтапного анализа. Данный метод основывается на использовании на первом этапе (предварительный анализ сетевого трафика) легковесной модели, которая позволяет исключить значительную часть нормального сетевого трафика, и на втором этапе (уточнение класса сетевой атаки) – более тяжелой модели (с точки зрения затрачиваемых вычислительных ресурсов), которая позволяет более точно классифицировать отобранные на первом этапе записи. В ходе экспериментального исследования была выбрана следующая комбинация моделей: Decision Tree и Random Forest.

The paper presents a method of network attacks detection using two-stage analysis. This method is based on the use of a lightweight model in the first stage (prior analysis of network traffic) that allows to exclude a significant part of the normal network traffic and using a heavier model (in terms of computational resources) in the second stage (clarifying the class of the network attack) that allows to more accurately make classification of the records selected in the first stage. In the course of the experimental study, the following combination of models was chosen: Decision Tree and Random Forest.

Введение

На сегодняшний день все больше компаний внедряют сетевые технологии в свои бизнес-процессы, что позволяет упросить как внутренний документооборот, так и взаимодействие с клиентами, однако порождает угрозу совершения атак на внешнюю (доступную из сети Интернет) и внутреннюю (не доступную из внешней сети) сетевую инфраструктуру. Согласно отчету Check Point Research [1] существует тенденция ежегодного увеличения числа сетевых атак, совершаемых на организации по всему миру.

В данной работе также рассматривается проблема, затронутая в ходе предыдущего исследования [2] – проблема обнаружения сетевых атак в смешанных сетях. Смешанные сети в контексте данной работы представляют собой единовременное присутствие в организации как классической, так и сегментированной сетевой архитектуры. Такое состояние сети возможно в ходе процесса сегментирования корпоративной сети, который, согласно докладу Akamai [3], может занимать от двух и более лет.

Для выявления сетевых атак в организации используются NIDS (от англ. Network-based Intrusion Detection System), данный класс систем можно разделить на anomaly-based (выявление аномального поведения, которое отклоняется от стандартного) и signature-based (выявления известных сигнатур, которые соотносятся с сетевыми атаками). Так как anomaly-based NIDS позволяют выявлять новые сетевые атаки, что является необходимостью при условии постоянного совершенствования методов и инструментов, используемых злоумышленниками, то данный класс NIDS обрел большую популярность при организации безопасности корпоративных сетей. Anomaly-based NIDS являются очень чувствительными к характеристикам сетевого трафика, используемым при обучении моделей, лежащих в их основе, что делает их зависимыми от присутствующей в организации сетевой архитектуры, а следовательно, делает невозможным их применение в контексте смешанных сетей.

Таким образом, проблема обнаружения сетевых атак в смешанных сетях, сопряженная как с усложнением характера сетевого трафика, так и с необходимостью исключать характеристики сетевого трафика, зависимые от сетевой архитектуры [2], в совокупности с проблемой увеличения числа сетевых атак на организации порождает необходимость разработки метода обнаружения сетевых атак, способного функционировать независимо от присутствующей в организации сетевой архитектуры.

Анализ предметной области

Системы обнаружения вторжений

Системы обнаружения вторжений Intrusion Detection Systems (далее IDS) представляют собой средства (представленные программным и/или аппаратным компонентами), предназначенные для мониторинга активности в компьютерных системах и сетях с целью выявления попыток несанкционированного доступа, атак или других аномальных действий.

IDS можно классифицировать по двум основным критериям: методу обнаружения и месту внедрения.

Классификация по методу обнаружения:

– signature-based IDS: выявляет вторжения, сравнивая сетевой трафик или события с заранее известными шаблонами атак. Такие IDS эффективны против известных угроз, но бессильны перед ранее неизвестными атаками (zero-day);

– anomaly-based IDS: выявляет отклонения от нормального (или эталонного) поведения. Подобные отклонения классифицируются как аномалии. Преимуществом такого подхода является возможность обнаружения новых, ранее неизвестных угроз. Основной недостаток – высокая вероятность ложных срабатываний (в данном случае имеют место как ошибки первого рода (ложноположительное срабатывание), что порождает необходимость SOC (с англ. Security Operations Center) тратить время на анализ заведомо недостоверного события, так и ошибки второго рода (ложноотрицательные ошибки), что приводит к опасной ситуации, когда пропускается потенциальная атака);

– гибридные IDS: совмещают оба вышеуказанных подхода, пытаясь уравновесить точность обнаружения и количество ошибок первого и второго рода. Такой подход наиболее распространен в крупных организациях, однако ключевым недостатком является высокая цена внедрения, что обусловлено необходимостью разворачивать в сети организации как signature-based, так и anomaly-based IDS.

Классификация по месту внедрения:

– HIDS (с англ. Host-based IDS): устанавливается непосредственно на оконечное устройство (сервер или персональный компьютер) и анализирует события операционной системы (далее ОС), системные журналы и файлы (по принципу функционирования частично дублирует функциональности, которые присутствуют у антивирусного программного обеспечения (далее ПО));

– NIDS: размещается в сети организации (может быть подключено как последовательно между сетевыми устройствами (коммутаторы и маршрутизаторы), так и размещаться в виде отдельного сервера, запрашивающего данные о сетевом трафике с сетевых устройств), и анализирует сетевой трафик в режиме реального времени.

В рамках данной работы внимание акцентируется именно на anomaly-based NIDS. Рассматривая данный класс систем более подробно, можно выделить следующие методы, используемые при их реализации:

– статистический анализ: данная группа методов позволяет определить отклонение поведения от нормального на основании расчета статистических характеристик сетевого трафика и сравнения этих значений с теми, что были определены при штатном функционировании сети организации. Данные методы могут использовать распределение вероятностей, средние значения и стандартные отклонения;

– методы машинного обучения: данная группа методов включает в себя множество подгрупп (например, методы обучения с учителем и обучения без учителя и т. п.), каждая из которых может быть применена в зависимости от контекста задачи и характеристики исходных данных (например, при недостатке размеченных данных имеет смысл применять методы обучения без учителя);

– методы глубокого обучения: данная группа методов схожа с методами машинного обучения, однако в ее основе лежит применение нейронных сетей, что в большинстве случаев позволяет получить более точный результат классификации за счет увеличения затрат вычислительных ресурсов.

Методы обнаружения сетевых атак

Purushotham P. и др. [4] в своей работе представили метод обнаружения сетевых атак, основанных на применении улучшенной модели Random Forest. Улучшение заключается в применении полусогласовательного (с англ. semi-voting) подхода, в рамках которого определяется 50 % наиболее надежных деревьев, голоса которых учитываются при вынесении окончательного решения. При оценке данного метода с использованием набора данных CICIDS2017 был получен результат, равный 0.938 (согласно F1-score).

Al-zubidi A. F. и др. [5] в рамках своей работы продемонстрировали метод противодействия DoS (с англ. denial-of-service) и DDoS (с англ. distributed DoS) атакам, основанный на использовании гибридной модели глубокого обучения. Данный гибрид включает в себя применение CNN (с англ. Convolutional Neural Network) и LSTM (с англ. Long Short-Term Memory) на этапе отбора наиболее значимых признаков, а XGBoost – в качестве классификатора. Был получен F1-score, равный 0.992 при оценке на наборе данных CICIDS2017.

Khan F. A. и др. [6] в ходе своей работы рассмотрели применение методов машинного обучения для многоклассовой и бинарной классификации при использовании метода аугментации данных SMOTE-Tomek. Наилучший результат при бинарной классификации был получен для метода Random Forest (F1-score = 0.997, набор данных CICIDS2017).

Djama A. и др. [7] представили метод обнаружения DDoS атак, использующий метод гибридного машинного обучения. Гибридная модель представляет собой объединение сети радиальных базисных функций (с англ. Radial Basis Function Networks (далее RBFN)) с методом SVM (с англ. Support Vector Machines), оптимизированным за счет использования алгоритма кластеризации K-means. При оценке данного метода с использованием CICIDS2017 был получен F1-score, равный 0.990.

Wu Z. и др. [8] в своей статье представили NIDS, основанную на модели Transformer, которая использует позиционное кодирование и механизм самовнимания (с англ. self-attention) для выявления аномалий в сетевом трафике. Применение данной модели позволяет сократить размерность данных без потери важных признаков и показывает высокую точность при использовании наборов данных, подверженных классовому дисбалансу. В результате оценки этого подхода был получен результат, равный 0.992 (согласно F1-score, набор данных – CICIDS2017).

Tang T. A. и др. [9] в рамках своей работы представили метод обнаружения сетевых атак, применимый для SDN-сетей (с англ. Software-Defined Networking). Данный метод использует рекуррентную нейронную сеть (с англ. Recurrent Neural Network (далее RNN)) на основе блоков GRU (с англ. Gated Recurrent Unit). При использовании CICIDS2017 был получен F1-score, равный 0.990.

Li Z. и др. [10] в своей работе представили гибридную нейронную сеть на базе LSTM и MSCNN (с англ. Multi-Scale CNN) для обнаружения сетевых атак. Также в данной работе используется аугментация данных с помощью VAE (с англ. variational auto-encoder) и GAN (с англ. generative adversarial networks). В ходе оценки предложенного метода были получены следующие значения F1-score: 0.837 (набор данных – NSL-KDD) и 0.989 (набор данных – AWID).

Mohammad R. и др. [11] в своей работе рассмотрели вопрос повышения точности обнаружения сетевых атак при применении моделей глубокого обучения (в этой работе рассматриваются 5 архитектур на базе CNN) с помощью использования при обучении синтетических данных. Авторы используют метод аугментации SMOTE для генерации синтетических данных на основании исходной выборки. Для набора данных CICIDS2017 наилучшая оценка была получена при использовании пятой рассмотренной в работе архитектуры для CNN и, согласно F1-score, она составляет 0.990.

Balakrishna T. K. и др. [12] представили модификацию метода LSTM в контексте задачи обнаружения сетевых атак. Модификация заключается в применении алгоритма оптимизации BWO (с англ. Black Widow Optimization). Для оценки данного метода использовался набор данных CICDDoS2019 (F1-score = 0.994).

Yangyang L. и др. [13] представили метод обнаружения сетевых атак, основанный на XGBoost, используемый для классификации сетевых атак, и MRMR (с англ. Maximum Relevance Minimum Redundancy) для выбора наиболее релевантных признаков. В данном методе также используется модификация SMOTE, которая заключается в применении K-means кластеризации (k-means-SMOTE). Для оценки разработанного метода применяется собственный набор данных, полученная оценка F1-score – 0.961.

Анализируя представленные в данном обзоре работы, можно сделать несколько наблюдений:

– наибольшей популярностью пользуются методы на базе машинного и глубокого обучения;

– при реализации гибридных методов основной акцент производится на точности обнаружения при игнорировании потенциального возрастания вычислительной нагрузки;

– имеет место применение методов аугментации данных при работе с наборами данных, содержащих сетевой трафик.

Таким образом, подавляющая часть существующих на сегодняшний день методов обнаружения сетевых атак игнорируют возрастание вычислительной нагрузки при применении более точных подходов к анализу сетевого трафика, что делает такие методы неприменимыми в контексте обнаружения сетевых атак в высоконагруженных сетях крупных организаций, что порождает необходимость разработки метода обнаружения сетевых атак, позволяющего применять тяжеловесные модели машинного обучения при минимизации времени, затрачиваемого на анализ данных.

Метод обнаружения сетевых атак с использованием двухэтапного анализа

Метод обнаружения сетевых атак включает в себя следующие этапы:

– первый этап: первичный анализ сетевого трафика, суть которого сводится к сокращению объема данных, которые необходимо проанализировать на следующем этапе. Для этой цели необходимо использовать легковесную модель, которая сможет обработать весь массив данных за относительно короткий промежуток времени;

– второй этап: вторичный анализ записей, которые на первом этапе были отмечены как аномальные. За счет сокращения объема данных к обработке на данном этапе можно использовать более затратную с точки зрения вычислительных ресурсов модель, которая позволит выполнить более точную оценку.

Представленный метод также включает в себя этап предобработки сетевого трафика, который включает в себя следующие шаги:

– исключение характеристик сетевого трафика, подверженных влиянию (согласно результату, полученному в ходе предыдущей работы, к таким характеристикам относятся: количество сетевых пакетов, объем сетевого трафика, статистические характеристики, связанные с количеством сетевых пакетов или объемом сетевого трафика (например, среднее длин сетевых пакетов или соотношение протоколов передачи данных TCP/UDP));

– инициализация коэффициентов, используемых при тонкой настройке гиперпараметров метода аугментации данных;

– тонкая настройка гиперпараметров метода аугментации (производится в течение 5 итераций и включает в себя следующие действия: генерация синтетических экземпляров данных с использованием метода аугментации данных SMOTE; расчет метрик валидации синтетических данных с использованием теста Колмогорова – Смирнова (оценка различия между оригинальными данными и сгенерированными) и оценки переобучения модели (с использованием ROC-curve оценивается разница в точности при обучении на тренировочном наборе данных (включает оригинальные и синтетические экземпляры) и обучении на тестовом наборе данных (включает только оригинальные экземпляры)); обновление гиперпараметров SMOTE с использованием ранее рассчитанных значений коэффициентов и метрик);

– применение метода аугментации данных SMOTE с использованием полученных в ходе тонкой настройки гиперпараметров.

После предобработки данные используются для обучения и оценки моделей, используемых в рамках двухэтапного анализа сетевых атак.

Экспериментальное исследование

Выбор и разработка программных инструментов

С целью оценки предложенного метода была выпол-

Листинг 1. Функция train_and_evaluate класса SMOTEOptimizer

```
class SMOTEOptimizer:
    ... - Реализация этапа предобработки сетевого трафика

    def train_and_evaluate(self, X, y, optimized_params, my_model, model_name):
        smote = SMOTE(**optimized_params, random_state=1337)
        X_aug, y_aug = smote.fit_resample(X, y)
        X_train_aug, X_test_aug, y_train_aug, y_test_aug =
          train_test_split(X_aug, y_aug, test_size=0.25, random_state=1337)
        now = datetime.now()
        my_model.fit(X_aug, y_aug)
        my_model_predict = my_model.predict(X_aug)
        index_1 = [index for index in range(len(my_model_predict)) if
          my_model_predict[index] == 1.0]
        other_X_aug, other_y_aug = X_aug.loc[index_1], y_aug.loc[index_1]
        other_model = RandomForestClassifier(n_estimators=100,
          random_state=1337)
        other_model.fit(other_X_aug, other_y_aug)
        other_model_predict = other_model.predict(other_X_aug)
        pos = 0
        for index in index_1:
            my_model_predict[index] = other_model_predict[pos]
            pos += 1
        f1_aug = f1_score(y_aug, my_model_predict)
        print(f"\t[{model_name}] F1-score: {f1_aug:.4f}, Time = {datetime.now()
          - now}")
```

нена его программная реализация с использованием языка программирования Python, а также следующих модулей: numpy, pandas, scikit-learn, imbalanced-learn.

Программный код предлагаемого решения представлен в листинге 1.

Вызов функции train_and_evaluate производился из функции run, программный код которой представлен в листинге 2.

Эксперимент

1. Используемый набор данных

В этой работе использовался набор данных CICIDS2017 [14] (если быть более точным, использовалась выборка Friday-WorkingHours-Afternoon-PortScan).

Этот набор данных имеет следующие характеристики:

– количество записей: 286096;
– количество полей: 79;
– количество классов: 2 (BENIGN и PortScan).

2. Оценка методов машинного обучения при индивидуальном использовании

В рамках данного пункта были выбраны следующие модели машинного обучения для оценки точности обнаружения (согласно метрике качества F1-score) и времени затрачиваемого на обучение и тестирование модели:

– Logistic Regression;
– Naive Bayes;
– Decision Tree;
– KNN (с англ. K-Nearest Neighbors);
– Random Forest;
– MLP (с англ. Multi-layer Perceptron).

Результат оценки представлен в таблице 1.

Согласно результатам, представленным в таблице 1, рассмотренные методы можно разделить на основании времени, затрачиваемого на обучение и тестирование. Таким образом, были выделены легковесные (Logistic Regression, Naive Bayes, Decision Tree, KNN) и тяжеловесные (Random Forest, MLP) модели машинного обучения.

3. Оценка методов машинного обучения при двухэтапном анализе сетевого трафика

В рамках данного пункта было рассмотрено совместное применение легковесных и тяжеловесных методов машинного обучения при двухэтапном анализе сетевого трафика.

Отобранные в ходе предыдущего этапа тестирования легковесные модели (Logistic Regression, Naive Bayes, Decision Tree, KNN) поочередно применялись вместе с моделями, отнесенными к группе тяжеловесных (Random Forest, MLP).

Результат совместного применения методов представлен в таблице 2.

Как можно увидеть по результатам, представленным в таблице 2, наилучшая комбинация по совокупности оценки F1-score и затрачиваемого на обучение и тестирование времени – использование Decision Tree для первичного анализа сетевого трафика и Random Forest для уточнения полученной на первом этапе оценки.

Сравнительный анализ

Сравнительный анализ предложенного метода с аналогичными решениями, направленными на обнаружение сетевых атак, представлен в таблице 3. Также в данной таблице приведен лучший (согласно F1-score) результат из таблицы 1, который использовался в качестве baseline (т. е. под данным обозначением представлен результат самостоятельного применения модели с присутствием предобработки данных, но без применения двухэтапного анализа). При выборе аналогичных решений предпочтение отдавалось методам, которые тестировались на исполь-

Листинг 2. Функция run

```
def run():
    light_models = {
        'Logistic Regression': LogisticRegression(max_iter=200),
        'Naive Bayes': GaussianNB(),
        'Decision Tree': DecisionTreeClassifier(),
        'KNN': KNeighborsClassifier(n_neighbors=3)
    }
    optimizer = SMOTEOptimizer(smote_hyperparams={'sampling_strategy': 0.5,
      'k_neighbors': 9})
    try:
        X, y = optimizer.load_dataset()
        X, y = balance_dataset(X, y)
    except NotImplementedError:
        exit(0)

    for model_name, my_model in light_models.items():
        best_params = optimizer.optimize_hyperparameters(X, y, my_model)
        optimizer.train_and_evaluate(X, y, best_params, my_model, model_name)
```

Таблица 1

Оценка методов машинного обучения при индивидуальном использовании

№	Модель	*F1-score*	Время
1	Logistic Regression	0.9199	0:00:04.044
2	Naive Bayes	0.6459	0:00:00.222
3	Decision Tree	0.8548	0:00:07.943
4	KNN	0.9047	0:00:18.186
5	Random Forest	0.9273	0:01:01.490
6	MLP	0.9331	0:02:42.674

Таблица 2

Оценка методов машинного обучения при двухэтапном анализе сетевого трафика

№	Модель	*F1-score*	Время
1	Logistic Regression + Random Forest	0.9111	0:00:14.928
2	Naive Bayes + Random Forest	0.9219	0:00:31.332
3	Decision Tree + Random Forest	0.9855	0:00:17.219
4	KNN + Random Forest	0.9353	0:01:28.473
5	Logistic Regression + MLP	0.9082	0:00:21.029
6	Naive Bayes + MLP	0.9084	0:00:46.915
7	Decision Tree + MLP	0.9731	0:00:23.497
8	KNN + MLP	0.9326	0:01:31.296

зуемом в данном экспериментальном исследовании наборе данных CICIDS2017, а оценка производилась с использованием F1-score.

Применение метода двухэтапного анализа сетевого трафика позволило не только улучшить результат, полученный при использовании методов машинного обучения индивидуально, но и получить результат, сопоставимый с аналогичными решениями, представленными в таблице 3.

Заключение

Предложенный в данной работе метод обнаружения сетевых атак с использованием двухэтапного анализа позволяет получить следующие преимущества:

– в результате исключения подверженных влиянию характеристик сетевого трафика метод становится независимым от присутствующей в организации сетевой архитектуры, что позволяет его использовать в классических, сегментированных и смешанных сетях;

– в результате применения аугментации данных решается проблема классового дисбаланса, которая является характерной для наборов данных, используемых в рамках задачи обнаружения сетевых атак;

– в результате применения двухэтапного анализа при обнаружении сетевых атак удалось значительно улучшить точность (согласно оценке F1-score) при незначительном увеличении времени (приблизительно 10 секунд), затрачиваемом на обучение и тестирование.

Также необходимо отметить, что применение двухэтапного анализа позволяет использовать при обнаружении сетевых атак тяжеловесные модели (например, Random Forest или MLP), которые на практике используются крайне редко ввиду высокой вычислительной нагрузки этих методов, недопустимой при необходимости обработки значительных объемов сетевого трафика.

В ходе оценки предложенный метод продемонстрировал результаты, сопоставимые с аналогичными решениями, а также позволил значительно улучшить

точность обнаружения сетевых атак относительно baseline (F1-score = 0.933). Таким образом, наилучший результат был получен в результате использования комбинации методов Decision Tree и Random Forest, при оценке с использованием CICIDS2017 было получено значение равное 0.985 (согласно F1-score).

В дальнейшем планируется продолжить исследование в области методов обнаружения сетевых атак с целью улучшения точности обнаружения сетевых атак при снижении вычислительной нагрузки.

Литература

1. Check Point Research Reports Highest Increase of Global Cyber Attacks seen in last two years – a 30 % Increase in Q2 2024 Global Cyber Attacks // Check Point Research : [Электронный ресурс]. – URL: https://blog.checkpoint.com/research/check-point-research-reports-highest-increase-of-global-cyber-attacks-seen-in-last-two-years-a-30-increase-in-q2-2024-global-cyber-attacks/ (дата обращения: 29.03.2024).

2. Метод предобработки трафика при выявлении сетевых атак в смешанных сетях / Н.Д. Колесников, Д.А. Устин, Д.А. Есипов, И.Ю. Попов // Информация и Космос. – 2024. – № 4. – С. 88–98.

3. The State of Segmentation 2023: Overcoming deployment obstacles proves to be transformational // Akamai : [Электронный ресурс]. – URL: https://www.akamai.com/resources/white-paper/2023-state-of-segmentation (дата обращения: 29.03.2024).

4. Purushotham, P. Classification of Cyberattack detection in Network Traffic using Machine learning techniques / P. Purushotham, A. Muddana // 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI). – 2024. – Vol. 2. – P. 1–6.

5. Al-zubidi, A. F. Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model / A.F. Al-zubidi, A.K. Farhan, S.M. Towfek // Journal of Intelligent Systems. – 2024. – Vol. 33, No. 1. – P. 1–24.

6. Balanced Multi-Class Network Intrusion Detection Using Machine Learning / F.A. Khan, N. Alshammry, A.A. Shah [et al.] // IEEE Access. – 2024. – Vol. 12. – P. 178222–178236.

7. Djama, A. Hybrid Machine Learning Approaches for Classification DDoS Attack 7 / A. Djama, M. Maazouz,

Таблица 3

Оценка точности разработанного метода согласно метрике качества F1-score

Метод	Модель	Набор данных	*F1-score*
baseline	MLP	CICIDS2017	0.933
Предложенный метод	Decision Tree + Random Forest	CICIDS2017	0.985
Purushotham P. и др. [4]	Random Forest	CICIDS2017	0.938
Al-zubidi A. F. и др. [5]	CNN + LSTM + XGBoost	CICIDS2017	0.992
Khan F. A. и др. [6]	Random Forest	CICIDS2017	0.997
Djama A. и др. [7]	RBFN + SVM + K-means	CICIDS2017	0.990
Wu Z. и др. [8]	Transformer	CICIDS2017	0.992
Tang T. A. и др. [9]	RNN	CICIDS2017	0.990
Li Z. и др. [10]	LSTM + MSCNN	NSL-KDD	0.837
		AWID	0.989
Mohammad R. и др. [11]	CNN (arch. 5)	CICIDS2017	0.990
Balakrishna T. K. и др. [12]	LSTM	CICDDoS2019	0.994
Yangyang L. и др. [13]	XGBoost	Собственный набор данных	0.961

H. Kheddar // 2024 1st International Conference on Electrical, Computer, Telecommunication and Energy Technologies (ECTE-Tech). – IEEE, 2024. – P. 1–7.

8. RTIDS: A robust transformer-based approach for intrusion detection system / Z. Wu, H. Zhang, P. Wang, Z. Sun // IEEE Access. – 2022. – Vol. 10. – P. 64375–64387.

9. Deep recurrent neural network for intrusion detection in SDN-based networks / T.A. Tang, L. Mhamdi, D. McLernon [et al.] // 2018 4th IEEE Conference on network softwarization and workshops (NetSoft). – IEEE, 2018. – P. 202–206.

10. Li, Z. An intrusion detection method combining variational auto-encoder and generative adversarial networks / Z. Li, C. Huang, W. Qiu // Computer Networks. – 2024. – Vol. 253, No. C. – P. 110724.

11. Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach / R. Mohammad, F. Saeed, A.A. Almazroi [et al.] // Systems. – 2024. – Vol. 12, No. 3. – P. 1–18.

12. Balakrishna, T. K. Multivariate Long Short-Term Memory with Spark Module for an Intrusion Detection System / T.K. Balakrishna, S. Sharma // International Journal of Intelligent Engineering & Systems. – 2025. – Vol. 18, No. 1. – P. 779–790.

13. A Network attack detection model of smart grid based on XGBoost algorithm / Y. Lian, L. Gao, P. Fang [et al.] // Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceeding of the 16th International Conference on IIHMSP in conjunction with the 13th international conference on FITAT. – 2021. – Vol. 2. – P. 481–488.

14. Network Intrusion dataset (CICIDS2017) // Kaggle : [Электронный ресурс]. – URL: https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset (дата обращения: 22.08.2025).